Communication Modeling in the Joint Integrated Mission Model (JIMM) and the Air Combat Environment Test & Evaluation Facility (ACETEF)

Michael D. Chapman and David W. Mutschler
<michael.d.chapman@navy.mil>
<david.mutschler@navy.mil>
Battlespace Modeling & Simulation Division (5.4.2)
Air Combat Environment Test & Evaluation Facility (ACETEF)
NAVAIR – NAWC-AD Patuxent River MD, USA

Abstract

The Joint Integrated Mission Model (JIMM) is a complex analytical model that provides the threat environment and main control for integrated operation of installed system test in the NAVAIR Air Combat Environment Test & Evaluation Facility (ACETEF). JIMM has a highly flexible mechanism for communication known as "user defined messages (UDMs)" that allows specific and detailed modeling of message execution, construction, composition, and content. However, because of this flexibility, messages do not possess a static format. Instead, they must be specifically translated given content and context into the required protocols and formats when interfaced with external stimulators. This paper will discuss communication modeling in JIMM, the translation of that modeling in ACETEF, and issues with its associated use in installed systems test.

Introduction

Originally intended as the merger of the Suppressor model and the Simulated Warfare Environment Generator (SWEG), the Joint Integrated Mission Model is a highly complex and flexible language-based model [LAT06]. Using its own text-based language known as the JIMM Conflict Language (JCL), analysts can create scenarios and include extensive data capture for detailed modeling of platforms and systems in multi-player engagements and campaigns. Applications using JIMM include human behavior modeling [LML06], weather modeling [KVSZ04], and radar system integration [Wor02]. In addition, JIMM allows the substitution of simulated systems with interfaces to real-time external systems. In this manner, the external system will act and react as if it were operating in the simulated world.

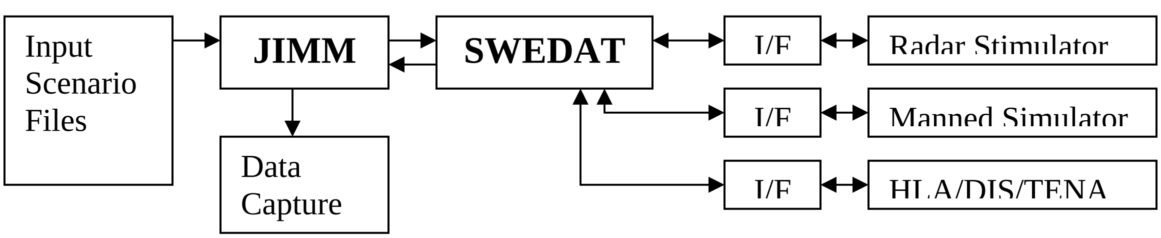

Figure 1 – Integrated Operation with JIMM & SWEDAT

For this reason, the Air Combat Environment Test & Evaluation Facility (ACETEF) uses JIMM as both a threat environment and as a main controller for real-time integrated operation [MOS02], [Mut05]. The medium of communication for integrated operation is a reflective shared memory protocol known as Simulated Warfare Environment Data

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **Communication Modeling in the Joint Integrated Mission Model (JIMM) and the Air Combat Environment Test & Evaluation Facility (ACETEF)**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **NAVAIR ? NAWC-AD,Battlespace Modeling & Simulation Division (5.4.2),Air Combat Environment Test & Evaluation Facility (ACETEF),Patuxent River,MD, 20670-154**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
2006 ITEA Modeling and Simulation Conference, Dec 11-14, 2006, Las Cruces, NM

14. ABSTRACT
The Joint Integrated Mission Model (JIMM) is a complex analytical model that provides the threat environment and main control for integrated operation of installed system test in the NAVAIR Air Combat Environment Test & Evaluation Facility (ACETEF). JIMM has a highly flexible mechanism for communication known as ?user defined messages (UDMs)? that allows specific and detailed modeling of message execution, construction composition, and content. However, because of this flexibility, messages do not possess a static format. Instead, they must be specifically translated given content and context into the required protocols and formats when interfaced with external stimulators. This paper will discuss communication modeling in JIMM, the translation of that modeling in ACETEF, and issues with its associated use in installed systems test.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**11**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Transfer (SWEDAT). SWEDAT is used to echo state information for players, platforms, and systems. In addition, a mailbox mechanism exists for the first-in-first-out (FIFO) exchange of messages between JIMM and the integrated interfaces (and optionally between the different interfaces themselves). Messages through shared memory are also known as "dispatches" and will be referred to as such in order to avoid confusion with simulated messages between simulated entities.

Like other system types, communication systems may be modeled in JIMM and executed constructively. They may also be controlled by external interfaces. In this manner, an outside process can participate in the exercise by sending and receiving messages. Furthermore, by using a given context, interfaces can inject additional detail (e.g. ordering of bits and explicit timing constraints) not available in the JIMM model.

This paper will first briefly discuss how communication and messages are modeling in JIMM. It will then discuss how interfaces and their corresponding external systems can interface in JIMM given these messages. Lastly, the paper will discuss how some interfaces have provided this additional detail as needed by their own interoperating environments.

Constructive Modeling in JIMM

Modeling of communication is JIMM is both extensive and flexible. It includes operation of communication transmitters and receivers, modeling of communication nets, selection of nets, general message types, and message information format given specific message types.

Message Types

JIMM has a very flexible mechanism for specification of message types in that the data transmitted in the messages must be specified explicitly via individual message data items (MDIs). In other words, message content is not implicit to the message type.

```
MESSAGE-NAME airborne
   A. target_info INFORMATION
      REGARDING: SENDER
      INCLUDE-DATA
         3D-POSITION    PITCH    HEADING      SPEED
         LOCAL-TRACK-ID  PLAYER-TYPE  PLATFORM-TYPE
         PERCEIVED-ELEMENT(S)       PERCEPTION-TIME
      END INCLUDE-DATA
END MESSAGE-NAME airborne
```

Figure 2 – Message Type Definition in JCL

Message types fall into two general categories: information and directives. Information messages provide data on players, either the sender or potential targets. This data may in turn have been acquired through sensors or by communication with other players. Information messages may also contain MDIs associated with "modes of control". Players may examine modes of control during execution of tactics to determine proper action against a target. A more flexible mechanism known as target actions or message actions may also be associated with the target perception. Unlike modes, target actions may be explicitly cancelled.

Directives are messages that must contain specific data. In other words, they must contain specific MDIs necessary to properly execute an associated action. Examples of directives include player creation (e.g. fill request) or for altering a perception of a target player's zone. MDIs associated with information and directives cannot be part of the same message.

Message may be organized by paragraph. However, paragraphs are really only provided as a convenience for scenario developers and do not mark separate messages transmitted simultaneously. However, they do provide a method for specifying multiple target actions.

Communication Nets

In JIMM, nets are uniquely defined by a user id and a net type. Net types are defined collectively for all players and specific instantiations are created for each scenario as the players are specified.

```
NET TYPE landline  MODE: INTERMITTENT  CHANGE FREQ DELAY: 13.5 (SEC)
    NO-SIGNAL-LEVEL-CALCULATION   USE GROUP: message_users
  MSG airborne           TRANSMIT-TIME:  2.54 (SEC) 1-WAY PRIORITY: 11
  MSG assignment_status  TRANSMIT-TIME:  2.96 (SEC) 1-WAY PRIORITY: 4
  MSG wpn_assign         TRANSMIT-TIME:  2.78 (SEC) 1-WAY PRIORITY: 6
  MSG wpn_assign_w_cuing TRANSMIT-TIME:  2.78 (SEC) 1-WAY PRIORITY: 6
  MSG cancel_assignment  TRANSMIT-TIME:  2.96 (SEC) 1-WAY PRIORITY: 4
  MSG intell_report      TRANSMIT-TIME:  4.57 (SEC) 1-WAY PRIORITY: 2
  MSG mode_ctrl_change   TRANSMIT-TIME:  4.57 (SEC) 1-WAY PRIORITY: 5
  MSG launch_request     TRANSMIT-TIME:  4.57 (SEC) 1-WAY PRIORITY: 5
  MSG alter_zone_msg     TRANSMIT-TIME:  2.34 (SEC) 1-WAY PRIORITY: 4
```

Figure 3 – Net Type Definition in JCL

Net types have two different modes: CONTINUOUS or INTERMITTENT. In continuous mode, all active transmitters on the net are continually transmitting and hence, may be detected by sensors that checking for emissions. With intermittent nets, a transmitter is only transmitting when actually sending a message. Messages are organized in groups. Each message type in that group has an associated transmission time for the net. Furthermore, messages are transmitted in order of highest priority. Messages with the same priority are transmitted in first-in-first-out (FIFO) order. All messages move from senders to receivers (i.e. "1-way"). Responses are handled as separate messages. Only one message is transmitted at a time.

Message Transmission may be modeling implicitly or explicitly. If the NO-SIGNAL-LEVEL-CALCULATION specification is used, messages will always reach their intended receivers. Otherwise, the option CALCULATE-SIGNAL-LEVEL can be specified, in which case a message transmission will fail if the signal to noise ratio (S/N) is low, the signal is jammed (a.k.a. non-lethal disruption), or the line of sight (LOS) is obstructed by terrain.

Transmitters and Receivers

In JIMM, though receivers can exist independently, transmitters are always linked with receivers and the pair is treated collectively. For this reason, transmitter specifications are associated with the receiver. The pair can be on, off, or non-operative. They are part of a single specific net. They start with an initial frequency and may move through alternate frequencies given jamming.

```
PLAYER: 13 cmd_post LEVEL: 1
  PLATFORM: 1 cmd_post_site X,Y,Z: 360.0  160.0 (KM)  0.0 (M)  AGL
    ELEMENT: 11  cmd_post_ele     DISCRETE QUANTITY: 3
      COMM-RCVR 112 comm_rcvr ON FREQ: 2.3 (GHZ) NET: 11 broadcast
        ALT-FREQ: 2  2.34 (GHZ)  ALT-FREQ: 4  2.38 (GHZ)
        ALT-FREQ: 7  2.39 (GHZ)  ALT-FREQ: 3  2.41 (GHZ)
    END ELEMENT
  END PLATFORM
END PLAYER
```

Figure 4 – Player, Comm. Receiver, and Net Instantiation in JCL

Transmitters and Receivers may also have associated encryption keys [Chap02b]. Systems not employing the same encryption key will not be able to interpret transmitted messages. Encryption keys may be changed during execution of tactics.

Determining Transmission of Messages

Whether or not message transmission is initiated is determined by the execution of tactics by players in a JIMM scenario. Some types such as LETHAL-ASSIGNMENT and INTELL-SEND are oriented toward sending messages once a specific resource (e.g. subordinate) is selected. However, other tactics such as reactive movement, weapon firing, and jamming can also be constructed to result in message transmission. This transmission is specified in JCL through the SEND-MESSAGE instruction.

```
INTELL-SEND normal_tactics
   PERCEPTION-TYPE ANYONE
      USE INPUT FOR FILTER 1
         PLAYER-TYPE close_sam_cdr
            INTENDED-RECIPIENT IS COMMANDER
               RE: CMD-CHAIN intell
            AND TIME-SINCE-LAST-INFO-SENT > 15.0 (SEC)
               RE: MESSAGE-TYPE intell_report
            AND HAS-DETECTED IS NO
 USE FILTER 1 SELECTIONS FOR FILTER 2
         PLAYER-TYPE close_sam_cdr
            3D-DIST < 35.0 (KM)
            AND SENSOR-DIRECT-SOURCE IS short_search/trk_rx
      FROM FILTER 2 SELECTIONS
         CHOOSE-FROM
            close_sam_cdr
               SEND-MESSAGE intell_report
                  REFER-TO THIS-TARGET RECIPIENT: THIS-PLAYER
            PICK-AT-MOST 99 NOW
END INTELL-SEND
```

Figure 5 – INTELL SEND Tactics in JCL

In general, the transmission of messages is specified through these tactics. However, to ease scenario development, some messages are initiated automatically such as during Identification Friend or Foe (IFF) interrogation and response [Chap02b].

Furthermore, messages are normally sent to a single receiver within a specified command chain. However, options in the SEND-MESSAGE instruction also allow messages to be multicast to all perceived players or broadcast to all players on a net [Chap02b].

Communication Net Selection Tactics

In JIMM, once a message is to be transmitted, a net for the transmission is selected using COMM-METHOD-SELECTION tactics. Selection is determined given different tactical criteria such as whether or not the net is busy, expected delay until transmission, shortest delivery time et al.

```
COMM-METHOD-SELECTION normal_tactics
   MSG-TYPE assignment_status weapon_assignment cancel_assignment
         intell_report mode_of_control_change
      USE INPUT FOR FILTER 1
         COMM-METHOD landline
            NET-CARRIES-MSG-TYPE IS YES
            AND RECIPIENT-ON-NET IS YES
      USE FILTER 1 SELECTIONS FOR FILTER 2
         COMM-METHOD landline
            NET-BUSY IS NO
      FROM FILTER 2 SELECTIONS
         CHOOSE-FROM
            landline
            PICK-AT-MOST 1 NOW
END COMM-METHOD-SELECTION
```

Figure 6 – JCL Communication Method Selection Tactics

In JCL, there is an assumption that a player will only be connected to one net of any specific type. There is no tactical criterion to distinguish nets of the same type given their unique integer net identifiers. A software change request (SCR) will be written and submitted to address this shortcoming.

Virtual Operation

In JIMM, integrated operation via SWEDAT is programmed in a scenario file known as the Configuration Data Base (CDB). Instructions for specific interfaces (also known as assets) allow for specific instructions regarding communication. For example, interfaces can send instructions to JIMM to turn transmitters on and off. Also, given appropriate instructions via CDB JCL, communications between players are checked before they would go on a net and may be echoed or routed to interfaces. Messages may be filtered given the net, message type, source, and destination. A routed message may in turn be blocked or modified per the interface requirements. Lastly, an interface may inject its own messages into the simulation.

```
ASSET: 67 MINICREW
   GLOBAL-STIMULI
      SENDER: THE 10 MASTER-MODEL
```

```
            ECHO-MESSAGE
               ON-NET: ALL          MESSAGE: ALL
               FROM: THE 53 ew/gci  TO: ALL

         SENDER: THE 10 MASTER-MODEL
            ECHO-MESSAGE
               ON-NET: ALL          MESSAGE: ALL
               FROM: ALL            TO: THE 53 ew/gci  $ was ALL
      END GLOBAL-STIMULI
   END ASSET
```

Figure 7 -- JCL CDB Instruction for Echoing Messages to an Interface

In JIMM, a message is implemented as a linked list of message data items (MDIs). When messages are provided to an interface, the messages are reorganized as arrays. Interfaces must examine the array and reconstruct the message's structure given that information. Message definitions are echoed to assets beforehand. When sending a message back to JIMM, the model will take the array and do the same.

<u>Communication Protocols</u>

When operating given explicit protocols, interfaces must translate the data provided by JIMM into and out of the specific "bit-wise" formats. In this manner, the interface can operate as a conduit between JIMM's unstructured message definitions and the stringent formats of real-world communications protocols. In this section, we will discuss the necessary means used to perform this translation in the context of two protocols implemented at ACETEF. Interfaces developed for external protocols must function in two directions. They must be capable of taking the JIMM internal message format and translating it to the protocol format for messages routed out of the model, and reverse the process for incoming protocol messages. .

The first protocol is Link 16, within the Joint Tactical Information Distribution System (JTIDS). Link 16 is a standardized set of messages used to provide Command and Control data within a Communications, Navigation and Identification (CNI) system. JTIDS is a joint-service system for providing secure, integrated CNI, and comprises the communications portion of Link 16. Each Link 16 message consists of a 35-bit header, followed by a message specific number of 75-bit data blocks (padded to 48-bits and 80-bits, respectively). Each data block consists of fixed length data elements in a specific order, determined by the message type [SISO06].

The second protocol is the Common Data Link (CDL), within the Enhanced IADS Messaging in a Simulation/Stimulation Environment (EIMSE). CDL is an unclassified representation of an actual Integrated Air Defense System (IADS) command and control protocol, developed by the 412th TW/EWR. EIMSE was the development of an interface between JIMM and the Joint Communications Simulator (JCS) for the Threat Simulator Investments Working Group (TSIWG). The JCS provided a means of generating complex RF signals in a simulated environment consisting of thousands of CNI emitters as modeled in JIMM [Chap02a]. Each message consists of a fixed length 56-bit data block followed by a 16-bit redundancy code. Each data block consists of fixed length fields in a specific order, determined by the message label [NAWC97].

Packing and unpacking data from the proper protocol format is a straightforward exercise in bit-wise manipulation of integer data. The real problem in creating these messages lies in determining the type of data available from the format and its relationship to the JIMM internal data. There are five types of data equivalences that can occur, addressing data, direct data correspondence, indirect data correspondence, action request and non-correspondence.

Addressing Data

The first case is addressing data, i.e. data used to identify the sender and/or recipient of a message. JIMM does not directly handle this as message data, but as a fundamental part of the Talk event. There are two methods to deal with this. The first, and the most prevalent, is to hardwire this data in the interface to a specific value. For example, many CDL messages contain the field TAN or Target Accounting Number. This is a 16-bit number, divided into a 9-bit track identification number and a 7-bit numerical representation of an alphabetical station address. The track identification number has a direct correspondence to the JIMM MDI LOCAL-TRACK-ID. The station address identifies the sender of the message. In this case, we can associate a specific station address with a specific player in the interface, referenced by the JIMM player global ID. Whenever the interface needs to generate a TAN, it will choose that station address. When a message comes in, the interface reverses the process to obtain the player global ID for the sender. The advantage of this method is that it requires no modification of the model. The disadvantage is that if the external address changes, the interface must be changed and recompiled.

The second method is to modify the JIMM code to permit it to keep track of the external address data solely for the purpose of communicating it to the interface. For example, JTIDS determines the routing of messages using a network-id for the sender, and a network participation group defined by a slot in the JTIDS frame. Since this data could change within the framework of the exercise for which the JTIDS interface was originally developed, the JIMM code was modified to allow the specification of this addressing data in the JIMM Scenario Database (SDB). The interface requests this data as an initialization dispatch from JIMM. It can then use the data to determine the appropriate routing address for outgoing messages, or determine the corresponding JIMM recipient player for incoming messages.

```
JTIDS-PROTOCOL   UNIT-ID <id number>
$      start stop  t/r   NPG   message type
       1     1     T     5     J2.2
       1     1     T     5     J13.2
       2     767   R     6     NONE
END JTIDS-PROTOCOL
```

Figure 8 -- JIMM Example Language for JTIDS Protocol

The advantage of this method is that if the external address changes, only the JIMM database needs to be changed, no code needs to be recompiled. The disadvantage is it does require modification of the JIMM source code, and it requires JIMM to keep track of data that it does not need for its own purposes.

Direct Data Correspondence

The next case is direct one-to-one correspondence between the protocol and internal data. This is when the JIMM reported data is identical in definition to that of the external protocol. In this case, all that is necessary is to handle unit of measure conversions between the two prior to translation to the bit-wise format. An example of this would be altitude, defined as the distance above the surface of the earth. In JIMM, this is reported through the MDI “ALTITUDE”, and is a real number measured in meters. In JTIDS, it occurs in the J3.2 Air Track message, in the J3.2I Data Element “ALTITUDE, 25 FT”, which is an integer representation of the altitude in 25-foot increments. In CDL, it occurs in the Label 5 Position Amplification message, in the field “ALT”, which is an integer representation of the altitude in 10-meter increments. The interface conversion algorithm is straightforward. First perform a unit conversion on the real number (meters to feet for JTIDS, and not necessary for CDL). Add half the altitude interval, to insure that we round to the nearest interval, and do an integer truncation for that interval. Finally, check that the resultant integer fits within the bits allowed by the format, and set it to the maximum if it exceeds the limit.

```
Int to25ft(float elev) {
   int temp = (int)((elev * (float)MtoF + 12.5);
   if(temp < 0) temp = 0;          // limit min elev
   temp = temp / 25;               // elevation / 25 ft
   if(temp > = 8190) temp = 8190; // limit max
   return temp;
}
```

Figure # -- Interface Conversion Routine: JIMM ALTITUDE to JTIDS “ALTITUDE, 25 FT”

Indirect Data Correspondence

The next case is indirect correspondence. Here, the data needed by the protocol does not correspond directly to JIMM reported data, but can be calculated from the data available. An example of this would be target velocity components, defined as the magnitude of a target’s velocity in a given coordinate system’s x and y components. This is used in the CDL Label 15/0010 GCI Additional Track Data message, in the fields “X VEL COMP” and “Y VEL COMP”. While JIMM does maintain an internal perception of the velocity vector for a moving platform, this data is not reported in a MDI. Instead, JIMM reports the speed and heading of the target in the MDI “SPD” and “HEADING”, respectively. As both CDL and JIMM share the same frame of reference (positive x east, positive y north), the interface requires only a simple geometrical calculation to obtain the desired data. Another form of indirect correspondence deals with data that JIMM does not report out through an MDI, but which is still available to an interface through SWEDAT. In this case, the necessary data is pulled from the appropriate SWEDAT block and used to calculate the data necessary for the protocol. In the work done for EIMSE and JTIDS, this was never necessary; however, the capability is available for use.

Action Request

The third case, action request, is probably the most difficult to deal with, as it requires work in both the interface and the JIMM scenario, and requires considerable thought into what is actually being modeled. In the real world, message data of this type generally instructs the recipient to take some action or use some resource for an action. To properly handle such a message, we must not only send a flag into the model indicating the action to be taken, but also insure that the model simulates the appropriate real world behavior for that flag. For example, the CDL Label 3 Air Track Position message has a field called "DROP ALERT" indicating if the recipient should consider this target for possible action or drop it from further consideration. In the EIMSE scenario, this message could easily be handled by associating an ACTION "alert" with this message to indicate that the recipient was to consider the target for action, or conversely, associate the CANCEL "alert" to drop the target from further consideration. Of itself, however, this is insufficient, as the model has no inherent conception of what "alert" means. What we are attempting to model is whether or not the recipient will consider a target as a threat to be engaged. This involves the Resource Allocation LETHAL-ENGAGE-QUEUE-ADD and LETHAL-ENGAGE-QUEUE-DROP logic. A player maintains a list of all platforms it perceives. However, it only considers engaging and firing against those that have passed the LETHAL-ENGAGE-QUEUE-ADD logic. Conversely, it will no longer consider a target for engagement if the LETHAL-ENGAGE-QUEUE-DROP logic is passed. This can be tested through a Tactical Criterion TARGET-ACTION IS or TARGET-ACTION IS-NOT "alert". If an Air Track Position message has been received with ACTION "alert", TARGET-ACTION IS will evaluate as true. If no such message has been received, or an Air Track Position message with CANCEL "alert" has been received, TARGET-ACTION IS-NOT will evaluate as true. Thus, by using TARGET-ACTION IS in LETHAL-ENGAGE-QUEUE-ADD, we can insure that a player will only consider a target as valid for engagement if it receives the Air Track Position message with ACTION "alert". By using TARGET-ACTION IS-NOT in LETHAL-ENGAGE-QUEUE-DROP, we insure that a player will no longer consider a target upon receipt of an Air Track Position message with CANCEL "action".

Non-Correspondence

The final case, non-correspondence of data, is the easiest to handle. Here, the data has no impact on the internal interactions of JIMM. This usually refers to activities that are not modeled within JIMM or to data that is not of importance for the test exercise. For this data, the interface can be hardwired to provide a default data value for messages from the model, and to ignore such data when generating a message going into the model.

Conclusion

JIMM is a flexible model that meets both the constructive requirements of the analytical community and, with the real-time capability provided via SWEDAT, the stringent requirements of the Test & Evaluation (T&E) community. Communication is effectively modeled. Moreover, with proper interface interaction, extensive interoperation given specific message formatting and protocols has been achieved.

JIMM is the property of the U.S. Government and is managed and maintained by the JIMM Model Management Office (JMMO). The JMMO is housed at Patuxent River MD within the Battlespace Modeling & Simulation Division (Code 542) of the Naval Air Warfare Center Aircraft Division (NAWC-AD) of the Naval Air Systems Command (NAVAIR). The JMMO maintains sole distribution rights for JIMM. The JMMO handles software trouble reports (STRs) and Software Change Requests (SCRs). The JMMO may be reached via electronic mail at <jmmo@navy.mil>.

References

[Chap02a] Chapman, Michael D. "EIMSE Final Report". TSWIG Contract # ATEC-02-905. May 2002

[Chap02b] Chapman, Michael D. "JIMM IFF/SIF Capabilities Improvement Software Design Document". JIMM Model Management Office. October 2002. Available from the JMMO via <jmmo@navy.mil>.

[KVSZ04] Kelly, Michael; Vick, Shon; Schloman, John; Zawada, Frank. "A Weather Service for Introducing Dynamic Attenuation Factors in the Joint Integrated Mission model (JIMM). Proceedings from the Simulation Interoperability Workshop. Simulation Interoperability Standards Organization. 04F-SIW-107, Fall 2004.

[LML06] Long, Gordon D; Mutschler, David W; Lohman Gary. "Human Behavior Modeling in the Joint Integrated Mission Model". Fall Simulation Interoperability Work, Orlando FL. September 2006. SIW-06F-063

[LAT06] Lattimore, Peter et al. "JIMM 2.6.1 and JIMM ACE 5.0 Users Guides". JIMM Model Management Office. 2006. Available from the JMMO via <jmmo@navy.mil>.

[MOS02] McMaster, John; O'Day, Stephen; Sarratt, Jessica. "Millenium Challenge 2002 Playback and Excursions in the JIMM Environment". ITEA Modeling and Simulation Conference, Las Cruces NM. 2003.

[Mut05] Mutschler, David W. "Improved Integrated Operation in the Joint Integrated Mission Model (JIMM) and the Simulated Warfare Environment Data Transfer (SWEDAT) Protocol". ITEA Modeling and Simulation Conference, Las Cruces NM, December 2005

[NAWC97] NAWC-AD EC-STIM. "Frame Structures and X-Ref Tables for the Common Data Link (CDL) based on Technical Interface Design Plan Message Formats and Procedures for the Common Data Link, Rev. 9.2". September 1997.

[SISO06] SISO Link 16 Production Development Group. "Simulation Interoperability Standard Organization (SISO) Standard for: Link 16 Simulations ". Simulation Interoperability Standards Organization SISO-STD-002-2006 May 2006.

[Wor02] Worsham, Richard. "Northrop Grumman Radar Simulation (AVSIM)". Proceedings of the 2002 IEEE Radar Conference. April 2002. pg 176-186

Author Biographies

Michael D. Chapman has worked as a developer for SWEG/JIMM for nearly ten years. He obtained a doctorate in High Energy Particle Physics from the College of William & Mary in 1992. He was the project lead for Contract # ATEC-02-905 of the Threat Simulator Investments Working Group (TSIWG) entitled "Enhanced IADS Messaging in a Simulated/Stimulated Environment (EIMSE)”. He currently serves as the JIMM Deputy Model Manager.

David W. Mutschler has worked for the U.S. Navy for more than twenty (20) years. He obtained a doctorate in Computer and Information Sciences from Temple University in 1998. He was the lead developer for the 14B53A Single Acoustics Signal Processor (SASP) Acoustics Trainer (SAT) Upgrade (versions 4.0, 4.1, and 4.1.1) and the principle investigator for Project #7 of the Forces Modeling & Simulation Computational Technology Area (FMS-7) for the Common High Performance Computing (HPC) Software Support Initiative (CHSSI) entitled “JIMM Parallelization using Stateless Action Ramification Events (STARE) and Cautious Optimistic Control (COC)”. He also served as JIMM Model Manager from July 2004 to February 2006.